An Approach to Autonomous Operations for Remote Mobile Robotic Exploration

Caroline M. Chouinard, Forest Fisher, Daniel M. Gaines, Tara A. Estlin, Steve R. Schaffer
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
Phone: 818-393-5352
Email: *firstname.lastname*@jpl.nasa.gov

Abstract— This paper presents arguments for a balanced approach to modelling and reasoning in an autonomous robotic system. The framework discussed utilizes both declarative and procedural modelling to define the domain, rules, and constraints of the system and also balances the use of deliberative and reactive reasoning during execution. This paper also details the implementations of such an approach on two research rovers and a simulated rover all in a Mars-like environment. Intelligent decision-making capabilities are shown in the context of several unforeseen events which require action. These events test the system's framework by requiring the system to handle uncertainty in state and resource estimations and in real-world execution. Future work which further enhances the idea of balanced reasoning is also discussed.

TABLE OF CONTENTS

1. INTRODUCTION TO AUTONOMOUS ROVER TASKS

Planetary rovers are the most practical tool for exploring the many surfaces of the solar system today. With manned missions far from being realized due to extreme costs and undesired risks, and satellites unable to reap the benefits of gathering on-the-ground surface data (ie. soil samples, rock images), rovers are at least one positive solution to the surface exploration dilemma. However, using rovers currently requires a team of scientists to plan activities for each day, maintain such a plan and control the movement of each rover throughout the life of a mission. This lack of autonomy can cause many undesired results.

Consider the Pathfinder mission and the use of the Mars rover, Sojourner; the time delay of communication between Earth and Mars at their closest points was roughly ten minutes [Mishkin et al. 1998] and ranged up to just under twenty one minutes. With current technology, this still holds true today. If a scientist on the ground spots an unforeseen edge or rock where the rover could get trapped or damaged, the visual data is already twenty minutes past. The scientist could react by sending a message to stop the rover. The message is sent and another twenty minutes passes before receipt. Obviously the time delay of communication between Earth and Mars and other solar system bodies is too great for reasonable human-to-rover interaction. Not only is rover time wasted, but the delay may cause unnecessary risk to the rover. Onboard autonomy can solve these problems and can also provide many advantages to a spacecraft and its mission.

An autonomous spacecraft must abide by the following:

1. Science objectives must be achieved to their greatest extent both during the initial creation of a plan or command sequence and also during the execution of such a plan.
2. An autonomous rover must be able to recover from unforeseen events, such as weather delays and inaccurate visual data and must repair any possible problems as they arise.
3. Also if the rover makes better progress than expected or

conditions change in the rover's favor, any possible new opportunities should be exploited.

4. Most importantly, these tasks must be handled without the need of constant human interaction.

Autonomy is beginning to make its mark in space and robotic exploration endeavors. The Continuous Activity Scheduling, Planning, Execution, and Replanning (CASPER) [Chien et al. 2000] planning software is one package that can be used to aid autonomous spacecraft control. CASPER is used to model a spacecraft's resources and states while also defining domain constraints and hardware functionality. The continuous planner generates a sequence of tasks and monitors the status of executing tasks. If unexpected or particular events occur, CASPER can react accordingly, as needed and as defined. CASPER will be used in the Autonomous Sciencecraft Experiment (ASE) flight experiment in which visual data of Earth's surface will be analyzed by an onboard science data analysis system. When a particularly interesting target is determined, the three satellite spacecraft mission will react autonomously to this "scientifically interesting" data by realigning the constellation to better capture images of the target on the next pass along their orbit. [Chien et al. 2001]

The Task Description Language (TDL) executive system [Simmons and Apfelbaum, 1998], which monitors specific task execution including all related subtasks, is also being researched for use in robotic and spacecraft autonomy. Robotic and other event-driven architectures must be able to handle events which can occur at any time, either consecutively or asynchronously. TDL manages each task or event separately and is ideal for such architectures. TDL is currently in use in the Federation of Intelligent Robotics Explorers (FIRE) Project in cooperation with NASA's Intelligent Systems Program, where a team of Mars rovers are coordinating task execution to achieve goals and recover from faults [Simmons et al. 2002].

The CASPER and TDL systems have been implemented on and proposed for several kinds of space and robotic craft, but this paper will focus on work done in 2001 using the CASPER planning software and TDL executive as parts of an autonomous system called CLEaR (Closed Loop Execution and Recovery) on two research rovers Rocky7 and Rocky8 and in a simulated rover environment called ROAMS (ROver Analysis, Modelling, and Simulation).

2. Deliberative and Reactive Reasoning

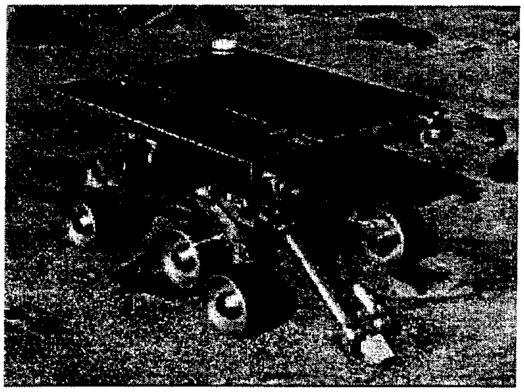
Consider a rover's day and the plan that controls what actions will occur throughout the day. A plan is a set of tasks and/or subtasks which is generated to realize a set of goals given by the user. Plan generation, and re-planning when conflicts arise, can be achieved using deliberation and/or reaction. A rover's day can be constrained both by tangible limitations on resources and from unforeseen events causing unpredictable problems. Resources such as energy and RAM as well as time or temporal constraints (per mission and per day) create obstacles which can be solved using a deliberative system, which searches the possible solutions given the projected system state. Deliberation is typically a better approach to solving conflicts in a plan when time is not a consideration. However, Murphy's Law dictates that uncertainty arises and can cause conflicts to the system state, so reactive behavior, which acts immediately to non-optimal progress, can be implemented as well. Reaction is a good method to solve a problem quickly, but an entirely reactive system does not have abstract knowledge of a domain and hence cannot recognize a global failure. The CLEaR concept [Fisher et al. 2002] balances both deliberative and reactive behavior to create a framework which can easily tackle many kinds of conflicts that arise. Throughout this paper "global" conflicts refer to errors that occur which require changes to future parts of the plan, while "local" conflicts are errors that require changes only to the currently executing part of the plan.. For example, an obstructed path may actually be a "dead end" that must be recognized globally and re-planned around and out of. The original plan could be completely re-ordered and activities may be added or deleted based on the new visual data. On the other hand, an obstructed path may also mean that the current path is no longer valid, but a small deviation would suffice to complete the current traverse. This would be a local conflict resolution that does not need to propagate up for a global domain fix.

Typically in the past, rover control systems have been a three tier structure made of a separate planner, executive, and control layer. The planner generates a plan and transfers tasks and domain knowledge to the executive and waits for response. The executive takes this information, processes or reprocesses the execution state through the functional layer, gives back the system state knowledge to the planner, and continues this cycle until the global state is successful. CLEaR collapses the top two tiers into one, combining the planner and executive to allow continuous updating of the system state. Instead of turning over the entire plan to the executive, only the current task is passed down from the planner and progress is monitored locally. This local control requires less interruption from the planner.

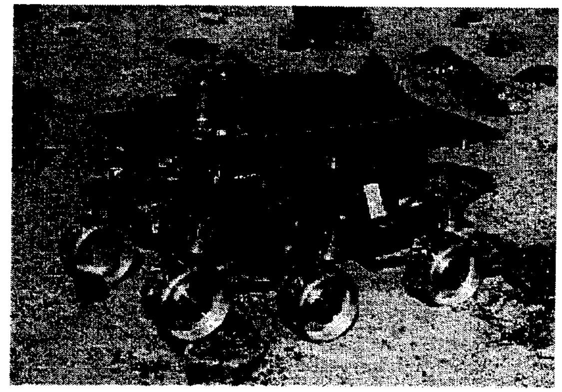
The CLEaR system is made up of a coupled interaction of deliberative and reactive behavior. The planner uses deliberation to find global solutions when necessary and to solve conflicts with the entire system state in mind. The executive using reactive reasoning to immediately solve local conflicts that may arise given that the solution at hand does not upset the system state.

3. Procedural and Declarative Modelling

CLEaR also balances the difference between procedural



A. – Rocky7



B. – Rocky8

Figure 1 - Rocky7 and Rocky8 Rovers

modeling and declarative modeling. A domain model combines the definitions of the spacecraft components with the functions that control them. Interdependencies can be structured so that any actions that need to be taken to resolve the dependencies between activities are defined. In a procedural control system, a task is broken down into the series of steps that will need to be taken to carry out an action. If an action is to be altered in any way, all interdependencies must be known and altered accordingly. In CLEaR, the software that controls the hardware components of the rovers, the functional layer, is a procedural control system. It describes how to carry actions out. Likewise, the TDL software component that issues and monitors tasks is a procedural system.

A declarative modelling system does not specify how to carry out tasks but instead, allows the user to specify what goals he or she would like to achieve without putting the pieces together of how to achieve such goals. Requirements and constraints are represented as are the relationships between events and objects. The CASPER planner is the generic reasoning system that makes up the declarative model in CLEaR.

Balancing the two modelling systems solves problems that both types, on their own, present. Procedural models are often too specific for use in generality, making them difficult to use or to adapt to work on many different spacecraft, while declarative models can become too computationally expensive to be considered for spacecraft given the limited resources of such bodies already stated. CLEaR's incorporation of the two modelling systems makes the representation of domain knowledge easier to write and use. See [Fisher et al. 2002] for more discussion on these tradeoffs.

4. THE CLEaR SYSTEM'S METHOD OF RESPONSE

In CLEaR, the planner is used to track state, resource, and time updates to enable anticipation of problems. The executive monitors local task progress and can solve immediate failures without disrupting global progress, when necessary. This type of interactive system provides full knowledge of rover constraints and can solve problems or conflicts and re-plan accordingly when new information changes a rover's originally scheduled plan. The planner serves as a database of knowledge which is constantly updated and can be queried for projected state values at any time. The executive monitors progress of activities and allows management of deviating tasks either by direct reaction, blind to the planner, or by passing control to the planner by failing an executing activity and instigating re-planning. Current resource usage knowledge combined with predicted or modelled usage allows more efficient response and repair by resolving conflicts before their "natural" end time.

The overall goal of CLEaR is to allow more science action per day and throughout the life of a rover's mission, which in turn provides more data to scientists. This is currently achieved by the following:

1. CLEaR allows a scenario to be re-planned without halting progress, and without entering a "failed" state. This saves time and keeps action the main focus.
2. With progress monitors in the executive, CLEaR can

Scenario Map for CLEaR in Action with Rocky7 and Rocky8

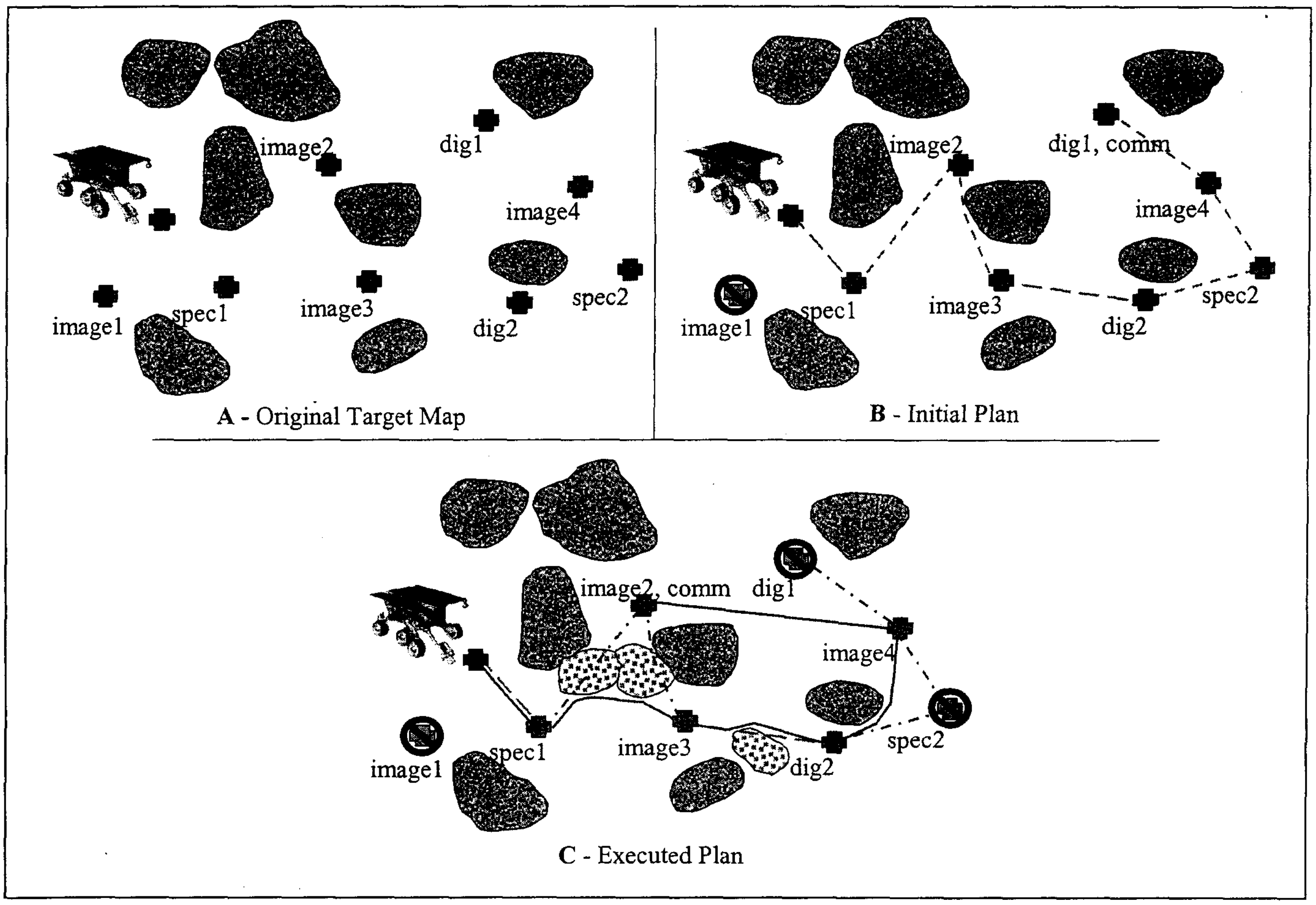


Figure 2 - Scenario Map for CLEaR in Action with Rocky7 and Rocky8

detect the need for re-planning earlier, which can decrease downtime and allow continuous progress.

The CLEaR team is also undergoing research to enable less deliberation by providing more information to the executive, which would allow less dependence on the planner for recovery. Less global re-planning means more action can be taken and more action means more data. As this is work currently in development, it will not be discussed in detail in this paper.

For more details on the CLEaR framework, see [Estlin et al. 2002] and [Fisher et al. 2002].

5. CLEaR in Action with Rocky7 and Rocky8

The rest of this paper will focus on two implementations of the CLEaR concept as well as explain some interesting work being done to improve the general CLEaR framework. The first example of the CLEaR concept in action is detailed by the CLEaR team working on the two research rovers Rocky7 and Rocky8. Rocky7, shown in Figure 1A., is similar in size and mass to the Mars Pathfinder mission's Sojourner rover. It is an all-wheel drive, partially-steered vehicle. Rocky8, shown in Figure 1B., is also and all-wheeled drive vehicle, but it is fully-steered and can therefore take advantage of different driving techniques, such as "crabbing", where all wheels turn to a particular position and the rover moves sideways without turning its body. Each of the rovers are controlled by an underlying robotic architecture which provides low-level control capabilities.

The scenario starts with nine science goals (including the start position) in JPL's Mars Yard. The targets consist of three different types of science activities (digs, images and spectrometer reads), and are each given a priority over or equal to each other. See Figure 2A. and 2B. for the rock distribution, target layout, and initially scheduled plan of these activities. The duration of the scenario is one day, however there is not enough time for all activities to complete by the end of the day. The CLEaR system will delete activities by priority as necessary. Several unforeseen events occur throughout the day which cause the planner or the executive to re-plan the scenario or manage the task, respectively. The first problem encountered is that of an unknown obstacle. Inaccurate visual data from descent imagery is simulated. One could also imagine that an image taken from one vantage point would not gather complete knowledge if a large rock were blocking the view of the area behind it. The scenario ran with obstacle avoidance which detected the blocked path. The active traverse activity failed and a reordering of goals resulted. The first obstructed path and the reordering of the scheduled plan is shown in Figure 2C.

The second unforeseen event occurs when an image is not fully compressed and there is no longer enough memory for all activities before the end-of-day communication download activity, which is the most important in the scenario. The planning system chooses to delete an activity over creating a new communication activity, because communication activities are too energy intensive and are only allowed at certain times of the day. Inserting a communication activity before the allowed time would violate the modelled temporal constraint. The deletion of an activity is done by priority. The third unforeseen event is similar to the second, in that an activity over-extends the energy resource. A dig is simulated to dig through soil that is tougher than expected resulting in a greater amount of energy being used. Thus, another activity deletion by priority is required. This deletion is shown in Figure 2C. as the deleted "spec2" activity.

The last unforeseen event, like the first, is also a result of inaccurate visual data, however at this obstruction, the executive monitors indicate that there is enough time and resources to complete the traverse, so the executive uses reactive reasoning to fix the traverse and complete without failure.

While this particular scenario does not exploit new opportunities, where an activity may be added during run-time, the CLEaR framework does support this behavior. Also while this scenario could conceivably be run without human interaction, some interaction was used to speed up time and to simulate the availability of new map data for use by the path planning algorithm. To resolve these issues, functionality has been added to automatically adjust the scheduled plan when activities run faster than modelled. Also, a new path planner is being used which can update its knowledge when new images are received from the hardware. Limited resources, temporal constraints and inevitable uncertainty were all factors CLEaR had to manage in this scenario. Rover energy and RAM were simulated on the rovers to be limited, time was limited to one day, and the visual data was simulated to be inaccurate.

To look back to the CLEaR concept's method of response, the planner takes model functions and parameters to determine how to create the different activities. Specific activity information is then added to create science activities where and when desired. Conflicts arise when resources are over-subscribed, usage of atomic hardware or other variables overlaps, and in general when constraints are not met. To create a plan, all conflicts must be repaired. CLEaR uses an iterative repair algorithm to resolve conflicts. From all the modelled information, the initial plan in the Rocky scenario is generated from one hundred forty three total conflicts. The planner receives updates at runtime and re-plans accordingly to unforeseen events. The executive manages progress throughout the day. During the first unforeseen event, the path deviates from the originally planned path so much that the failure is recognized prior to the end of the activity's time slot, and the planner is informed of the failure. The rover's current ending position is not congruent with the expected ending position of the traverse. The rover domain knowledge is updated, and the rover's position conflict is resolved. From this behavior, we see that the executive monitors allow the planner to fix problems before their "natural" time of failure. This creates "faster than now" response and repair and increases efficiency of the rover's time and resources.

The second obstructed path in the scenario uses reactive reasoning to resolve a local conflict. The planner is never told of a failed traverse; instead, the path of the rover is simply updated in the domain knowledge, but the traverse activity remains successful. This saves the system from ever entering a failed state and enables re-planning without halting progress.

The gains from the CLEaR concept are realized in the previously detailed scenario. Costly re-planning is minimized, executive monitors allow re-planning to occur earlier so that down time is virtually eliminated and continuous progress is enabled, and failed states are also minimized. All of these effects enable the rover to achieve more goals. The scenario was demonstrated in 2001 on both the Rocky7 and Rocky8 research rovers, using the exact same Decision Layer and with only the necessary changes to the Functional Layer to control the different hardware.

6. CLEaR in Action with ROAMS

The second scenario that the CLEaR system was implemented on was in cooperation with the ROAMS team. ROAMS is a high-fidelity simulation tool which has the capability to simulate a number of different rovers [Yen and Jain 1999]. See Figure 3. The scenario in this instance is much simpler than that of the real rovers. There are three goals, and every goal is reached without any conflicts. While the more complex features of CLEaR are not demonstrated in this scenario, the ability to switch out the actual rover is shown as another completely distinct CLEaR feature. Since the ROAMS scenario is not particularly complex, focus will be drawn instead to the expansion of current testing with ROAMS.

ROAMS Simulator

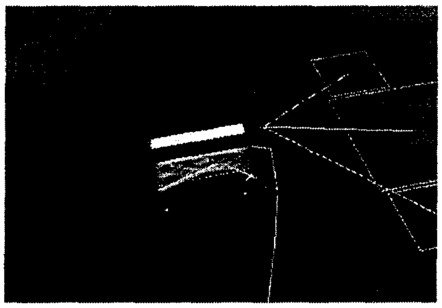


Figure 3 – ROAMS Simulator

With a well-defined command interface, CLEaR is able to communicate with any control layer that adheres to the API. ROAMS has been adapted to accept this command API. It also has several obstacle avoidance algorithms built in, so it is possible to run CLEaR with ROAMS and test different algorithms easily. Each algorithm causes deviating paths, much like what might happen on a real rover with drift. The CLEaR/ROAMS scenario is currently being expanded to take advantage of the CLEaR framework, and introduce activity failures and insufficient visual data. Opportunistic science, where new activities are added or previously deleted activities are re-introduced, is also being considered for future work with ROAMS.

7. Current Work in Flexible Time

Research is also being done with flexible time to make CLEaR run more efficiently. As is often the case, making certain assumptions about a problem can simplify the thought process. This is true for reasoning about actions.

Planning and more importantly re-planning with the use of a grounded time representation is much simpler than reasoning in the space of valid intervals. Although the use of such representation is much more limiting when it comes to the execution of the planned command sequence. For this reason we have begun working on a hybrid approach for reasoning (re-planning) in the space of grounded time, while executing in the space of flexible time.

Our first implementation towards this goal dynamically adjusts the start-times of actions during execution. The algorithm is:

1. Wait till there are no executing activities,
2. Pick the activity with the earliest start-time (but not yet executing),
3. Look to see if this activity can be placed earlier in the plan without creating any conflict to the plan.

As described this algorithm continuously tries to dynamically pack the leading the edge of the plan. While this is only a first step towards the more general problem, it does address some very important issues.

1. By only packing the leading edge we maintain *slack* within the schedule, which simplifies re-planning when and if it should occur.
2. By moving up the start-time of activities waiting to execute. The system reduces wasted downtime, while enabling the use of conservative/pessimistic estimates on duration.
3. By only packing during periods when no activity is executing and through the use of conservative time bounds on activity durations, the issue of how to push portions of the plan back as a result of activities taking longer than expected is addressed (in part).

This approach enables the use of conservative planning without the execution time inefficiencies. Further, because of the optimization capabilities of the CASPER continuous planner that our system is built on, as the dynamic packing of the plan occurs this will also allow for other activities (goals) that may have originally been omitted from the plan to be reintroduced, as time permits.

One significant limitation of the current approach is that activities can only be packed during idle periods in the plan. Our next step is to identify any non-executed activities in the plan which are not dependant on any steps not already complete (preconditions have already been met) and advance these activities' start-times. This would enable independent portions of a plan to be advanced independently.

Another approach we are working on is to use the underlying constraints, which are maintained in the plan structure, to map the plan back to a flexible time representation for the purposes of execution. This would enable the use of less conservative duration estimates while still addressing how to resolve the case of an activity completing later than anticipated.

8. Future Work

Two other areas of research that will affect the CLEaR framework are that of Atomic Resource Managers (ARMs) and Execution-Time Query (ETQ) capability. These related ideas allow even more control in the executive, making the executive reason deliberatively and act to some degree like a planner. Some preliminary testing has been done, and we are looking to permanently commit the new capabilities in the future.

The model of a rover defines resource usage per activity. However, activities can under or over-subscribe their allocated resources. Currently, when an activity is going to over-subscribe its resources, the activity is failed, and it is up to the planner to determine action by searching the entire domain for the proper repair of any conflicts. It is possible that the originally generated plan would not throw a conflict if an activity were to use more than its allocated amount of resources. ETQ allows the executive to query the planner for any such possible conflict information. When a conflict would not result, the executive invokes the appropriate changes to the plan and the activity's resource usages are extended. In this sense, ETQ allows the executive to make deliberative decisions, which are much faster than those that would be made by the planner.

ARMs enable the executive to maintain resources and "schedule" activities based on the intermittent availability of such resources. Some activities require atomic resources during execution, but only periodically. For example, during a navigation traverse, the onboard cameras may need to be used every 200cm. An image could take 5 seconds. The camera usage and traverse time in a navigation activity may be modelled to occur every 15 seconds, but the time it takes to actually make the traverse could vary from the model if the soil consistency varies or if there is an incline in the traverse. So, the cameras may not actually be used every 15 seconds. During the time of the traverse, it would be nice to fill the memory buffer with images, while the cameras are not is use. To do this, ARMs have been used. The process is to attach two activities together, in this case, a navigation command and an image command, and let the executive maintain knowledge of atomic resource usage (the cameras). The navigation activity lets the executive know that the cameras are available and the executive inserts an image activity whenever possible. ETQ at this point is also introduced to extend the paired activity to acquire as many images as possible until a conflict in the plan arises. The two concepts together are making the most efficient use of the rovers when working in uncertain environments.

9. CONCLUSIONS

CLEaR is a demonstration of balancing deliberative and reactive reasoning using procedural and declarative modelling. While motivation for these balances was previously discussed, the implementations of CLEaR with the different rovers prove the need for a combined approach and show success of several mission-like scenarios.

CLEaR's combination of its reasoning elements into one tier allows immediate reaction to failures and intermediary action from the planner when necessary. We have seen that this interaction of the planner and executive allows re-planning without halting progress, can limit the frequency of entering "failed" states, and can detect the need for re-planning earlier. These advantages save time, encourage more action and decrease downtime.

Without cooperative modelling, CLEaR would be susceptible to spacecraft-specific structured code, which would not be adaptable to different rovers, or also the CLEaR model could be too computationally expensive for the limited resources of rovers. Missions are highly constrained by time and resources, and it is most important to make most efficient use of a spacecraft's limiting factors. Scientists want the most and best data possible which is why autonomy and other AI systems are being used for rover control and data acquisition. Automating rovers with balanced reasoning and modelling is the clear approach.

10. RELATED WORK

Different types of spacecraft are using planners and executives to control and maintain their underlying architectures. The CASPER planner mentioned in this paper is being used in several missions with different executives, while at the same time, the TDL executive of CLEaR is being used in conjunction with other planners. The choice to use different planners and executives can be made based on many criteria, including but not limited to the proven competence of the system or language, availability of the package at the mission start, the desired or required level of control of the spacecraft, and of course the cost of the planner or executive.

The CASPER planning system is currently one part of the onboard autonomy system used by the Thee Corner Sat mission, which is a collaboration between NASA's JPL and the University of Colorado, Space Grant College [Chien et al. 2001]. The Three Corner Sat mission uses the Spacecraft Command Language (SCL) executive. The SCL has been and is being implemented in several other missions, including the FUSE satellite and the TechSat21 mission mentioned earlier (see www.sclrules.com). SCL which was designed to support spacecraft control, has been successful in many previous missions, however, TDL was chosen for the CLEaR work because it was specifically designed to support robotic task management.

The TDL executive has also been implemented on several projects with other planners, including the DIRA Project which uses a planner to coordinate the use of multiple robots [Smith et al. 2001]. The planner in the DIRA simulator must generate flexible hierarchical plans which allow the robots to cooperate without dependence on a stricter plan. Multiple-spacecraft missions require planners and executives to be more flexible to allow each of the rovers to handle tasks more efficiently and sometimes together.

Progress can be difficult to manage and plans can often change. Balancing the right modelling system with the most effective executive is a challenge with growing need for resolution, as more multi-robot missions are being considered. CASPER is currently in use with another multiple-robot coordination project called MISUS [Estlin et al. 1999].

The Remote Agent Experiment (RAX) (Jonsson, et al., 2000) was flown on the NASA Deep Space One (DS1) mission. It demonstrated the ability of an autonomy system to respond to high-level spacecraft goals by generating and executing plans onboard the spacecraft. The planner in RAX takes as input a schedule request and produces a flexible, temporal schedule for execution by its executive. A major limitation to this approach was that planning was only performed in a batch fashion. If re-planning was required, the spacecraft was "safed" until a new plan had been generated (which could be on the order of hours). Furthermore, since RAX was applied to a spacecraft, it did not handle issues with surface navigation and path planning.

ACKNOWLEDGEMENTS

This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

REFERENCES

[1] S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, D. Tran, "ASPEN – Automating Space Mission Operations using Automated Planning and Scheduling," *SpaceOps,* Toulouse, France, June 2000.

[2] S. Chien, B. Engelhardt, R. night, G. Rabideau, R. Sherwood, "Onboard Autonomy on the Three Corner Sat Mission," *Proceedings of the 2001 International Symposium on Artificial Intelligence, Robotics, and Automation for Space*, Montreal, Canada, June 2001.

[3] S. Chien, R. Sherwood, M. Burl, R. Knight, G. Rabideau, B. Engelhardt, A. Davies, P. Zetocha, R. Wainright, P. Klupar, P. Cappelaere, D. Surka, B. Williams, R. Greeley, V. Baker, J. Doan, "The Techsat-21 Autonomous Sciencecraft Constellation Demonstration," *Proceedings of iSAIRAS 2001,* Montreal, Canada, June 2001.

[4] T. Estlin, T. Mann, A. Gray, G. Rabideau, R. Castano, S. Chien, and E. Mjolsness, "An Integrated System for Multi-Rover Scientific Exploration," *Proceedings of AAAI-99,* 1999.

[5] T. Estlin, F. Fisher, D. Gaines, C. Chouinard, S. Schaffer, I. Nesnas, "Continuous Planning and Execution for an Autonomous Rover," *3rd International NASA Workshop on Planning and Scheduling for Space,* Houston, Texas, October 2002 (to appear).

[6] F. Fisher, T. Estlin, D. Gaines, S. Schaffer, C. Chouinard, R. Knight, "CLEaR: Closed Loop Execution and Recovery – A Framework for Unified Planning and Execution," Technology and Science IND News Issue 16, pg. 15-20, September 2002.

[7] F. Fisher, D. Gaines, T. Estlin, S. Schaffer, C. Chouinard, "CLEaR: A Framework for Balancing Deliberation and Reactive Control," *Proceedings of the AIPS On-line Planning and Scheduling Workshop,* Toulouse, France, April 2002.

[8] A. Mishkin, J. Morrison, T. Nguyen, H. Stone, B. Cooper, B. Wilcox, "Experiences with Operations and Autonomy of the Mars Pathfinder Microrover," *Proceedings of the 1998 IEEE Aerospace Conference,* March 21-28 1998, Snowmass at Aspen, Colorado.

[9] R. Simmons, T. Smith, M. B. Dias, D. Goldberg, D. Hershberger, A. Stentz, R. Zlot, "A Layered Architecture for Coordination of Mobile Robot," in Multi-Robot Systems: From Swarms to Intelligent Automata, A. Schultz and L. Parker (eds.), Kluwer, 2002.

[10] T. Smith, R. Simmons, S. Singh, and D. Hershberger, "Future Directions in Multi-Robot Autonomy and Planetary Surface Construction," *Proceedings of the 2001 Space Studies Institute Conference*, May, 2001.

[11] J. Yen and A. Jain, "ROAMS: Rover Analysis Modeling and Simulation Software," *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'99),* Noordwijk, Netherlands, June 1999.

Caroline Chouinard *is a software engineer at NASA's Jet Propulsion Laboratory. She develops software for rover domains on the CLEaR team in cooperation with several other JPL teams. She has enjoyed working in both the business software industry, specializing in AI related natural language processing tools, and also the scientific community, working on planning, rovers, and navigation algorithms. She received her BBA in International Business and her BA in Computer Science from the University of Texas at Austin.*